DOCUMENT RESUME

ED 284 448 FL 016 854

AUTHOR Wyatt, David H.
TITLE Applying Pedagogical Principles to CALL Courseware Development.
PUB DATE 82
NOTE 15p.; In: Smith, William Flint, Ed. Modern Media in Foreign Language Education: Theory and Implementation. The ACTFL Foreign Language Education Series; see FL 016 850.
PUB TYPE Information Analyses (070)

EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS Appropriate Technology; *Authoring Aids (Programing); *Computer Assisted Instruction; *Courseware; Educational Technology; Educational Theories; *Media Adaptation; Programing Languages; *Second Language Instruction; Teacher Role; *Teaching Methods

ABSTRACT

This paper on the application of principles to computer-assisted language learning (CALL) first clarifies the the relationships between the capabilities of the computer (and computer-controlled technology) and the main approaches to second language learning. Fourteen common types of CALL programs are briefly reviewed. A "relational" classification is proposed, based on the relationship between and roles assumed by learner and computer, to define instructional, collaborative, and facilitative types of programs. This tripartite classification is used as a basis for explaining why published reviews of CALL software are so often unintentionally misleading and why the possibilities for valid computer applications in some newer approaches to language learning have been almost completely overlooked. This classification is also used as a basis for discussing important aspects of CALL design and development, including educational authoring systems, educational programming languages, and general-purpose programming languages. (CB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☑ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

L Fiddle

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

ED284448

Applying Pedagogical Principles to CALL Courseware Development

David H. Wyatt
Specialized Curriculum Design Consultants and Trinity College

Introduction

In order to gain an accurate picture of the field of computer-assisted language learning (CALL), it is essential to make a crucial distinction between the medium itself and approach and methodology. It will be argued here that the computer medium is not tied to a particular type of pedagogy, but can potentially be used as a component of a very wide range of different approaches and methodologies. In recent years, increasing attention has been focused on second-language acquisition and natural approaches to second-language learning. As a medium for language learning, the computer may have as much potential for fostering acquisition and communication as more traditional tools.

Before examining the prospects for developing different types of CALL courseware, the relationships between the capabilities of the computer (and computer-controlled technology) and the main approaches to second-language learning will be clarified. Common types of CALL programs will be reviewed briefly under the familiar classification scheme (tutorial, drill-and-practice, etc.). A "relational" classification will then be proposed (based on the relationship between and roles assumed by learner

David H. Wyatt (M.A., Cambridge University) is the Director of Specialized Curriculum Design Consultants and a Lecturer in the Trinity College (Washington, D.C.) Masters Program for Computers in Education and Training. He has been instrumental in the installation of more than 1500 microcomputer systems for CALL. He is also Computer Editor for *System*, the author of *Computers and ESL* (Harcourt Brace Jovanovich/Center for Applied Linguistics), a member of ACTFL, TESOL, and CALICO, and past President of the Society for Microcomputer Applications in Language and Literature.

FL016854

BEST COPY AVAILABLE

and computer) to define (1) instructional, (2) collaborative, and (3) facilitative types of programs. It will be shown how this newer classification scheme lends insight into a wide range of issues in the use of computer software and hardware in language courses. This tripartite classification will also be used as a basis for explaining why published reviews of CALL software are so often unintentionally misleading, and why the possibilities for valid computer applications in some newer approaches to language learning have been almost completely overlooked. Finally, the new classification will be used as a basis for discussing important aspects of CALL design and development. Teacher involvement in the development and intelligent use of CALL is essential to its success; thus, this chapter will also indicate how the three major approaches to producing CALL programs allow for different types and levels of teacher participation.

CALL and Pedagogy

One of the most fundamental issues in the field of CALL is the "fit" between the computer's capabilities and the demands of language pedagogy. Neither is static; both have undergone very significant developments in the last decade, and continued change is to be expected. Nevertheless, it is illuminating to examine the question of "fit" in terms of currently accepted pedagogical principles and widely available microcomputer technology. Over the last ten years, microcomputers have become the first type of computer technology available to the language-teaching profession on a significant scale. Microcomputers at first appeared to have certain well-defined pedagogical capabilities (such as the presentation of mechanical language drills) and limitations (for example, an inability to handle open-ended student responses). As a result, much of the early CALL software was of the drill-and-practice variety, and microcomputers rapidly became identified with just one approach to language pedagogy—the audiolingual method. The introduction of an earlier technology—the tape recorder—should have alerted teachers to the danger of leaping to this type of conclusion. In the case of the language laboratory, the obvious "fit" between the technology and language learning appeared to lie in the provision of speaking practice. It was only later that the value of this kind of speaking practice was questioned and more valid possibilities for the development of listening comprehension were recognized for the medium.

Unfortunately, much of the debate over the use of computers in language learning has been based on their unilateral identification with mechanical audiolingual techniques. Drill-and-practice CALL materials of this type have been welcomed by some writers (Davies, 1; Wyatt, 13) but strongly criticized (Sanders and Kenner, 11; Underwood, 12) or

rejected (Krashen, 7) by others. Whatever the merits of this type of software, it is important to realize that CALL is not limited to and should not be identified with any single approach and methodology. In short, computer applications in second-language learning are potentially at least as relevant to the natural approach as they are in an audiolingual or cognitive code setting.

Types of CALL Programs

The discussion of computers and language learning that follows is limited to the largest class of computer applications—programs intended for *direct use* by students. Other applications are outside the focus of this discussion, as is the use of the computer and computer-controlled technology by the teacher as a demonstration medium in the classroom. A basic computer system can be used as a powerful "electronic blackboard," particularly in the teaching of reading and writing skills, provided that a large enough display device is available. With the addition of a computer-controlled videodisc or videotape player, a new spectrum of video-assisted language teaching (VALT) techniques become possible (Wyatt, 17). Two other significant areas beyond the scope of this chapter are computer-assisted language testing (Wyatt, 15) and teacher support (Higgins and Johns, 4).

Software for direct student use includes a bewildering variety of forms. Figure 1 shows some of the most common "types," but does not attempt to provide a complete listing. Related accounts of software types may be found in Davies (1) and Higgins and Johns (4).

Program Type	Examples of Functions and Contents
1. Tutorial	introducing new material—e.g., the Cyrillic alphabet in beginning Russian
2. Drill and practice	allowing mastery of material already presented—e.g., grammatical forms, culturally appropriate behavior
3. Game	adding elements of peer competition, scoring, and timing to a wide variety of practice activities
4. Holistic practice	providing higher-level, contextualized practice activities—e.g., cloze passages
5. Modeling	demonstrating how to perform a language task—e.g., how a good reader handles difficult sections of a reading passage
6. Discovery	providing situations in which linguistic generalizations can be made—e.g., inferring rules for generating comparative forms
7. Simulation	allowing students to experiment with language use—e.g., levels of formality in a conversational simulator

8.	Adventure reading (interactive fiction)	offering "participatory" reading materials—e.g., student as detective explores murder location, gathers clues
9.	Annotation	providing a wide range of language "notes" (vocabulary, syntax, plot, etc.) available on demand during reading or listening activities
10.	Idea processor	planning and editing outlines—e.g., before writing activities, after listening to lectures
11.	Word processor	creating and editing written assignments
12.	On-line thesaurus	expanding vocabulary, improving writing style
13.	Spelling checker	guarding against errors during or after writing activities
14.	Textual analysis	revealing structural and stylistic aspects of written work—e.g., complexity and variety of sentence types, subject/verb agreement errors

Figure 1. Some Types of Programs Used in CALL

Much discussion of CALL software has been conducted on the basis of categories such as those in Figure 1. Unfortunately, these categories can be very misleading. Software in a "game" format, for example, often consists of disguised drill-and-practice materials. "Tutorials" are frequently taken to imply mechanical approaches to language learning, whereas tutorial CALL formats can equally be used to implement functional and communicative language learning.

More insight into CALL approaches can be provided by examining the interactional relationship between student and computer. On this basis, three fundamental categories of language-learning programs can be distinguished: instructional, collaborative, and facilitative (Wyatt, 16).

In *instructional* CALL programs, the computer presents language-learning materials in a highly structured, predetermined manner. In essence, the student plays the relatively passive role of responder and the computer functions as an authoritative instructor. The language-learning objectives of instructional CALL programs can be very precisely specified, as can the learning path or paths to be followed by students in attaining the goals. Tutorial, drill and practice, holistic practice, and many types of game software (items 1 through 4 in Figure 1) are examples of instructional programs.

Despite their high degree of structuring, instructional programs need not appear rigid or mechanical. Students may be given considerable freedom to select program formats, move on to new programs, and the like. Based on their needs as demonstrated in their responses at the computer, students may be routed in very different ways through an instructional program and may receive individualized learning materials not presented

to other students. Nevertheless, instructional programs are generally characterized by highly predetermined sets of learning goals and paths, regardless of whether this is apparent to the learner.

Collaborative software casts students in a much more active role and depends on them to take responsibility for their own learning. The computer may function in a large range of roles—as an interactive world in which the student plays the part of detective or adventurer, as an environment for linguistic discovery, or as a conversational partner, etc. In all cases, however, the student must make decisions and take the initiative in working with the program.

In some types of collaborative programs (for example, adventure reading), learning objectives can be expressed only in very general, holistic terms, while in other cases (such as annotation and discovery courseware) they may be specified in as much detail as in instructional programs. One key feature of collaborative programs, however, is that there is no attempt to create predetermined learning paths. Students will interact with these programs and draw on their pedagogical resources in a wide variety of ways. Items 5 through 9 in Figure 1 are examples of collaborative program formats.

Facilitative software differs from the above two categories in that it is essentially empty of direct pedagogical content. With purely facilitative materials, the program itself embodies no language-learning paths or objectives. Rather, the software serves as a tool in language-learning activities. In Figure 1, items 10 through 14 are examples of programs that can be used facilitatively in second-language learning. Of these, word processing programs are the most familiar, already finding considerable use in writing activities where the ability to edit and rework materials quickly and easily is important to the learning process. This use of word processing illustrates the main function of facilitative software—to minimize the "inauthentic labor" (Kemmis et al., 6) involved in the learning process.

The three-way relational classification of program types, summarized in Figure 2, clarifies a number of aspects of CALL.

Approach	Characteristics
A. INSTRUCTIONAL e.g., tutorial, drill and practice, holistic practice, many "games"	• Students are responders, not initiators, despite their high level of activity • Detailed set of high- and low-level learning objectives • Predetermined learning path(s) • The computer instructs the student; students learn *from* the computer

B. COLLABORATIVE e.g., modeling, discovery, simulation, adventure reading, annotation, some "games"	• Students are initiators, take more responsibility for their learning • May only be possible to specify learning objectives in high-level terms • No predetermined learning paths • Elements of discovery learning; students learn *with* the computer
C. FACILITATIVE e.g., word and idea processing, spell check, on-line thesaurus, text analysis	• Students are initiators, entirely responsible for their learning • Learning objectives and paths not specified or embodied in computer program • Students use computer as *tool* to reduce "inauthentic labor"

Figure 2. Relational Classification of CALL Approaches

Instructional approaches in CALL represent the older traditions of computer-assisted learning. At their best, they are extremely sophisticated elaborations of the proven techniques of programmed instruction. The more precisely the learning objectives can be specified and the learning paths predicted, the more effective this type of software is likely to be. Researchers have generally reported significant reductions in average student learning times in language arts areas with instructional courseware (Edwards et al., 2; Ragosta et al., 8). A recent study (Robinson, 10) suggests that the computer program's ability to assess individual student's problem areas and provide practice in them is effective in second-language learning. However, much research remains to be done to assess whether the results in language arts can be replicated in appropriate areas of second-language learning, and this type of research is fraught with problems (Hope et al., 5; Pederson, Chapter 5 in this volume).

The strengths of instructional CALL software include a high degree of student involvement and activity, student self-pacing, practice materials appropriate in quantity and challenge to each student's level, and other aspects of individualization. In general, these benefits can only be realized if each student works at a separate computer. As a result, a relatively large number of computers is usually a *sine qua non* for the appropriate use of instructional software.

To achieve and maintain computer-mediated individualization, detailed recordkeeping on each student's work is often a feature of instructional programs. For greatest convenience and maximum ease of use, this requires an additional layer of computer technology to network the individual computers and provide centralized information storage. In general, instructional CALL therefore implies more expensive and sophisticated computer systems than other approaches.

It should be apparent that much of the debate concerning the use of

computers in language learning has been conducted as though CALL were synonymous with instructional software. Historically, this was perhaps an understandable error—much of the earlier CALL software was instructional, with particularly strong ties to the audiolingual and cognitive code approaches. In the mid-eighties, however, it seems high time to set instructional software in its proper context as just one of three distinct approaches to CALL.

It is also important to recognize that instructional software can serve as part of many communicative approaches to second-language learning. Newer instructional programs employing communicative approaches are already appearing. Moreover, some proponents of the natural approach advocate the provision of grammar-oriented materials for out-of-class study (Krashen, 7); in these circumstances, well-designed instructional software offers all the advantages previously outlined.

On the other hand, there may be little or no "fit" between some communicative approaches and instructional CALL, and this should be acknowledged openly. It serves little purpose for a reviewer opposed to the use of drill techniques to attack the use of computers in language learning based on a detailed critique of drill and practice and other instructional programs. Presumably the same reviewer would also reject a majority of the textbooks currently used in language learning because of the prevalence of drill activities; it is the methodology, not the medium, that generally lies at the heart of such debates. (See Chapter 9 in this volume, by Hubbard, for further discussion of software review and evaluation.)

Collaborative CALL provides a sharp contrast to the instructional approach. In all collaborative activities, the prime linguistic focus is on process, function, and communication. It is the students who decide how the pedagogical resources of the program are to be used.

Some important types of collaborative software are designed for use by groups of students. Group-oriented collaborative programs are fundamentally different from the single-student approach typical of instructional programs. In many CALL simulations, games, and discovery activities, for example, the student group discussions that precede each new "move" at the computer are one of the most valuable aspects of the program.

Certain types of collaborative software are primarily intended for single-student use. Reading skill modelers and annotated listening programs are examples of single-student collaborative software that takes advantage of the computer's ability to individualize the learning process. However, there are two crucial differences from the way the individualizing capabilities are used in instructional CALL. In these collaborative programs, the reading or listening activity proceeds transparently—without apparent computer intervention—until students decide to ask for assistance or check their understanding. Secondly, when help is requested,

it is the student who selects from and decides how to use the range of learning aids provided.

Since many collaborative program formats are group-oriented, this approach to CALL can be implemented even when relatively few microcomputers are available. Because many group-oriented approaches derive their value from student interactions rather than from explicitly programmed learning sequences, a careful selection of existing games and simulations for native speakers may be directly usable in the language classroom. Basic microcomputer systems are adequate for most applications, as recordkeeping features are generally absent from this type of software. Thus, group-oriented collaborative CALL may be the most practical starting point for a majority of teachers and institutions.

In their purest form, programs used in facilitative CALL embody no assumptions about language teaching and learning. As examples, consider the use of word processing and spelling-checker programs in language courses. It is now widely accepted that writing should generally be treated as a process, not a product (Raimes, 9), and so students' written assignments should go through at least one cycle of editing and rewriting. Similarly, research indicates that teacher responses to students' written work should emphasize more global problems such as planning, rhetorical structure, and content (Griffin, 3) rather than focus only on form. With handwritten or typed student assignments, rewriting can be laborious at each stage. When assignments are word-processed, however, even global changes can be incorporated quickly into successive drafts. Spelling checkers, now becoming standard components of many word processor programs, can both remove a burden and focus attention on problem items.

There are many existing microcomputer programs in foreign language as well as English versions that can serve effectively in CALL for purely facilitative applications in the teaching of writing. Such programs have no built-in pedagogical bias, so they are adaptable to any approach to second-language learning. Usable programs are often available for even the least sophisticated microcomputer systems. The only drawback is the need for a great deal of available computer time. Obviously, straightforward work on writing assignments in the classroom requires one computer per student. Even if the word processing is done out of class in a campus computer laboratory, students must still have ready access to computers for relatively lengthy periods. As institutions and students continue to acquire microcomputers at a rapid pace, however, this constraint on facilitative applications is likely to diminish.

A trend that tends to mask the identity of facilitative approaches is the combining of instructional and collaborative elements with a basically facilitative program. For example, software (such as Quill) is beginning to appear that attempts to integrate instructional programs on planning a writing assignment with word processing. In debating the merits of such

combined programs, the reviewer must be careful to specify which elements of the software are being discussed vis à vis their purpose and overall objective.

In summary, the three-way classification of CALL applications may provide a new perspective in the debate over computer applications in second-language learning. The appropriateness of the computer in a given approach may vary widely according to the category of CALL. In particular, there may be little or no "fit" between instructional CALL and some communicative approaches; on the other hand, there may be considerable scope in theory and practice for agreement between communicative approaches and collaborative and facilitative CALL.

Illustration of CALL Approaches to Pedagogy

The three main roles for computers and computer-controlled technology can be illustrated by considering a few of the possibilities for teaching listening comprehension via CALL. In this regard, one extremely promising strategy involves the use of a videodisc player under the control of a simple microcomputer system (Wyatt, 17).

For an instructional approach, the video materials generally need to be custom scripted and filmed so as to be fully integrated with the computer-based activities. The computer programs might present a careful sequence of listening exercises designed to develop various aspects of listening comprehension skills. Videodisc operation is under computer control throughout and presents the listening segments on which each student activity and items are based. Clearly, in an instructional approach of this type, the development of computer–video materials is expensive and time-consuming.

In some types of collaborative approaches, such as annotated listening, learning materials are based on existing video materials such as news broadcasts, documentaries, and movies. Computer programs are developed to provide a wide variety of assistance to students, ranging from minimal listening comprehension hints and vocabulary glosses to the provision of complete scripts. After beginning an activity, students listen uninterruptedly to the video materials until help is needed. At any point, students can touch a computer key to halt the video presentation and select the kind of collaborative assistance they want. Computer help is keyed directly to the current position in the video material, since the computer keeps track of the videodisc's progress. Such collaborative materials are relatively quick and inexpensive to develop for those abreast of the technology.

Existing or custom-filmed video materials can be used in a facilitative approach. Here, the computer simply functions as an intelligent video controller. The program knows where each scene or topic starts and ends,

providing students with rapid, convenient control over the video materials. Such a system also provides an ideal classroom aid for the teacher. A "template" computer program, once developed, would serve both teacher and student applications with minimal time required for adaptation to each new videodisc.

CALL Program Design and Development

The following paragraphs focus on two related considerations: how computer software is developed, and which developmental methods are appropriate for the different approaches to CALL. The discussion is based on the types of computer hardware and software generally available now and in the immediate future. Much more sophisticated technology is already in existence; unfortunately, its cost will keep it beyond the reach of virtually all educational institutions for a number of years.

The three main methods of producing CALL programs are through the use of (1) educational authoring systems, (2) educational programming languages, and (3) general-purpose programming languages (Wyatt, 14). In terms of current technology, each method has different implications for the role of the language teacher in the development process and the types of CALL programs that can readily be designed and produced.

Educational Authoring Systems

Educational authoring systems offer the greatest opportunity for teachers to develop their own CALL materials. The main purpose of these software systems is to permit the creation of relatively sophisticated student programs while requiring a minimum of learning and production time for the developer. With an authoring system, materials developers are not required to learn how to program a computer or deal with an operating system. Instead, they need only master the rules of the system, which are designed specifically for rapid learning and educational applications. In principle, teachers with no previous computer programming experience should quickly be able to begin developing their own CALL materials. In practice, however, the performance of many authoring systems is far from the ideal. Of the large number the writer has reviewed personally, well over half suffer from very poor user design. In many cases, assistance from someone with computer programming expertise would be necessary to operate even very "simple" systems. Other systems are riddled with programming "bugs" or contain surprising built-in limitations to their stated capabilities.

Many authoring systems are designed solely for the production of

instructional types of CALL materials. Some are placed onto a single-screen format, as with many cloze exercise generators. Others employ a question-and-answer structure that, used imaginatively, can produce a large number of superficially different screen formats and activities, all of which are fundamentally drill-and-practice in nature. If the instructional approach is acceptable for some purposes, however, these very limited systems may well be worthy of consideration. Some of the simpler systems are indeed easy to learn and will operate without flaws.

There are still other authoring systems that transcend these limitations. For example, both teachers and students can readily develop creative reading activities with certain authoring systems. A different type of collaborative CALL material can be produced with adventure authoring systems, although these are generally more difficult to master. A small number of very powerful authoring systems will permit much greater flexibility, as described below, but involve greater expense, complexity, learning periods, and development time.

Educational Programming Languages

Educational programming languages such as PILOT constitute the second developmental route. This method generally offers significantly greater flexibility in the type of CALL materials that can be produced. Potentially, both collaborative and instructional materials can be developed, although facilitative programs are beyond the capabilities of most educational programming languages.

As the term implies, the use of educational programming languages requires a knowledge of programming. In the original version of PILOT, for example, the programming language was designed to be extremely simple, so that learning to program basic (instructional) student exercises would take only a few hours. Thus, educational programming languages such as PILOT were intended to provide teachers with a quick entry point into developing computer-assisted learning materials. Unfortunately, the original simplified programming languages quickly earned the reputation of being unsuitable for producing worthwhile courseware. The emphasis is now on power and sophistication, not simplicity, and recent versions of PILOT reflect the trend toward full-featured educational programming languages with certain authoring-system capabilities. Languages such as EnBASIC and SuperPILOT are not intended to provide quick access for nonprogramming teachers. They are powerful, time-saving programming tools meant for relatively expert programmers.

In most cases, developing collaborative or instructional CALL materials using educational programming languages therefore implies a team approach involving at least one language teacher and one programmer. If possible, however, there should be at least two language teachers on the

team. One should have familiarity with CALL applications and some programming experience, while the other should have as few preconceived ideas about CALL as possible and little or no computer experience.

As noted previously, however, there is considerable application for existing programs (originally written to entertain or educate native speakers) in collaborative CALL. Carefully chosen programs such as simulations, adventures, and games may be usable directly, without any need for reprogramming. Teacher-prepared handouts for use before, during, or after the work on the computer can enhance the value of the software for second-language learning. In this manner, nonprogramming teachers may be able to develop excellent collaborative activities on their own.

General-Purpose Programming Languages

The remaining method of producing CALL materials is through the use of general-purpose programming languages (BASIC, Pascal, Assembler, and the like). In essence, these languages offer unrestricted access to the computer's full capabilities. Virtually any type of CALL application—instructional, collaborative, or facilitative—can be produced. Except for those willing to undertake considerable study of programming, however, language teachers will need to form teams with programmers to produce materials using this method.

General-purpose programming languages are the only realistic option for the development of original facilitative programs. Where possible, however, it is important to consider the alternative—adapting carefully selected existing programs, through teacher planning and handouts, for facilitative CALL. Word processors are a case in point. "Educational" word processing programs have been appearing, designed specifically for the classroom and, in a few cases, even for foreign language courses. Such programs aim at ease of learning and may have special capabilities designed for the classroom environment. However, they may sacrifice so much power and speed as to represent a poor bargain, especially when compared with a full-featured professional program carefully selected for user-friendliness. Because of the complexity of facilitative software development, the language teacher's energy and expertise can generally be best spent in choosing the most suitable of the extant programs and developing imaginative ways of using them.

Conclusion

This chapter has provided arguments for challenging two prevalent views

of CALL. One is the misconception that computers fit into only one pedagogical style and approach to language learning, a misconception based on the characteristics of instructional CALL. As computers become more available to students and teachers, their value as language-learning aids in collaborative and facilitative activities will be more widely recognized.

The second has challenged the assumption that language teachers wishing to become involved in CALL materials production must turn themselves into programmers when, in fact, nonprogramming teachers have a particular value as members of software development teams. Nonprogrammers can also produce effective CALL programs with carefully chosen authoring systems. Finally, there is considerable room for imaginative adaptations of existing non-CALL programs for the second-language classroom.

References. Applying Pedagogical Principles to CALL Courseware Development

1. Davies, Graham. *Computers, Language and Language Learning.* London: Centre for Information on Language Teaching and Research, 1982.
2. Edwards, Judith, Shirley Norton, Sandra Taylor, Martha Weiss, and Ralph Dusseldorp. "How Effective Is CAI? A Review of the Research." *Educational Leadership* 33, 2 (1975):147–53.
3. Griffin, C. "Theory of Responding to Student Writing: The State of the Art." *College Composition and Communication* 33, 3 (1982):296–301.
4. Higgins, John, and Tim Johns. *Computers in Language Learning.* Reading, MA: Addison-Wesley, 1984.
5. Hope, Geoffrey R., Heimy F. Taylor, and James P. Pusack. *Using Computers in Teaching Foreign Languages.* Language and Education: Theory and Practice, no. 57. Orlando, FL: Harcourt Brace Jovanovich/CAL, 1984.
6. Kemmis, Stephen, Roderick Atkin, and Eleanor Wright. *How Do Students Learn? Working Papers on Computer-Assisted Learning: UNCAL Evaluation Studies.* Norwich, England: Centre for Applied Research in Education, 1977.
7. Krashen, Stephen. "The Power of Reading." Plenary presentation at the annual TESOL Conference, New York, 1985.
8. Ragosta, Marjorie, Paul Holland, and Dean Jamieson. *Computer-Assisted Instruction and Compensatory Education: The ETS/LAUSD Study.* Washington, DC: U.S. Department of Education, 1982.
9. Raimes, Ann. "What Unskilled ESL Students Do as They Write: A Classroom Study of Composing." *TESOL Quarterly* 19, 2 (1985):229–58.
10. Robinson, Gail, John Underwood, Wilga Rivers, José Hernández, Carollyn Rudesill, and Clare Malnik Enseñat. "Computer-Assisted Instruction in Foreign Language Education: A Comparison of the Effectiveness of Different Methodologies and Different Forms of Error Correction." San Francisco: Center for Language & Crosscultural Skills, 1985. [EDRS: ED 262 626.]
11. Sanders, David, and Roger Kenner. "Whither CAI? The Need for Communicative Courseware." *System* 11, 1 (1983):33–39.
12. Underwood, John. *Linguistics, Computers, and the Language Teacher: A Communicative Approach.* Rowley, MA: Newbury House, 1984.
13. Wyatt, David. "Computer-Assisted Language Instruction: Present State and Future Prospects." *System* 11, 1 (1983):3–11.

14. _____. "Three Major Approaches to Developing Computer-Assisted Language Learning Materials for Microcomputers." *CALICO Journal* 1, 2 (1983):34–38.
15. _____. Computer-Assisted Teaching and Testing of Reading and Listening. *Foreign Language Annals* 17, 4 (1984):393–407.
16. _____. *Computers and ESL.* Orlando, FL: Harcourt Brace Jovanovich/CAL, 1984.
17. _____. "ESL Applications of the Computer-Controlled Videodisc Player." *Computers and the Humanities* 18, 3–4 (1984):243–49.